SSC San Diego Biennial Review 2001

1

Next-Generation Information Systems

■

1 NEXT-GENERATION INFORMATION SYSTEMS

C^4ISR Imperatives—Cornerstones of a Network-Centric Architecture

Clancy Fuzak, William L. Carper, Mary Gmitruk, James W. Aitkenhead, Tom Mattoon, and Victor J. Monteleon
SSC San Diego

ABSTRACT

Network-centric operations are military operations that fully exploit the availability of "universal" connectivity. Much discussion of network-centric operations focuses on envisioning future applications of the connectivity. These future applications are a confederation of pieces, not a single unit. The prerequisite for fielding these pieces is an in-place network-centric architecture that can support their implementation. SSC San Diego has identified seven command, control, communications, computers, intelligence, surveillance, and reconnaissance (C^4ISR) imperatives that represent command capabilities needed by military forces. Network-centric architecture requires effectively achieving five of these imperatives. This paper argues the importance of these five, and suggests the value of building technologies to enable these imperatives. This approach allows clearer understanding of the application of technology while assuring consistency with the end objective of network-centric operations.

INTRODUCTION

Network-centric operations have been the focus of serious discussion over the past several years, especially following the wide exposure provided by Admiral Cebrowski's 1998 *U.S. Naval Institute Proceedings* article [1]. Here we take the view that network-centric operations are military operations that fully exploit the availability of "universal" connectivity. Such connectivity can lead to:

- Widespread access to heretofore isolated resources (people, machines, data)
- Improved access to specialized information that has, in the past, been difficult to locate
- Accelerated planning processes
- Introduction of a new dimension to "contact" between opposing forces—cyber contact
- Innovative uses of information
- Development of entirely new ways to work and to think about tasks
- Emergent operational concepts and organizational structures
- Et cetera—think, for example, about emerging Web services and Web uses for personal or business reasons

There will no doubt be many innovative applications for the future network as we build toward network-centric operations. Much discussion of network-centric operations focuses on envisioning these future applications—most of which have not yet been invented. These applications are a confederation of pieces, not a single unit. In fact, that is an intention—the ability to evolve and adapt through "parts upgrade," without having to replace an entire system. The prerequisite for fielding these pieces is an in-place network-centric architecture that can support their implementation. And as is the case with the Web, applications follow infrastructure. Make access simple and widespread, make providing content relatively easy, and someone invents eBay. In this view, "network-centric architecture" provides ubiquitous and universal, timely and "useful" access.

IMPERATIVES FOR C^4ISR

SSC San Diego has identified a set of seven command, control, communications, computers, intelligence, surveillance, and reconnaissance (C^4ISR) imperatives. These imperatives represent command capabilities that have been needed by military forces throughout history and are expected to continue to be needed in the future. While the imperatives are time-independent, the degree to which they can be achieved depends upon available technology.

Dynamic Interoperable Connectivity will provide assured, user-transparent connectivity, on demand, to any desired locations in the "infosphere"—the worldwide grid of people, sensors, military databases, fusion nodes, national resources, and commercial and other non-U.S. information resources.

Universal Information Access will use that connectivity to access strategically located sensors, database servers, and anchor desks. It will provide users, at all levels, with the key information needed to create and share a consistent perception of the operational situation.

Focused Sensing and Data Collection provides the warfighter with the ability to acquire the information needed to allow viewing an area of interest or responsibility at any desired level of fidelity and resolution.

Achieving *Consistent Situation Representation* is the fourth imperative. When all key operational commanders have a consistent situation understanding, tools supporting the fifth imperative, *Distributed Collaboration*, can be used to work effectively together across space and time to plan and execute missions and tasks.

The sixth imperative, *Information Operations–Assurance*, will protect our information and our C^{4}ISR infrastructure.

Finally, *Resource Planning and Management* provides the mechanisms for effective use of all available resources.

Implementing a network-centric architecture requires effectively achieving several of these imperatives.

NETWORK-CENTRIC ARCHITECTURE

The concept of "ubiquitous and universal, timely and 'useful' access" needs some discussion. The first point we should make is that "access" does not equal information access, which we will discuss later as the imperative for Universal Information Access. In the network-centric architecture, access implies the ability to establish relationships among users. Those relationships must support the users' timeliness requirements. The users might be people, or processes running on machines. Examples of access might be one person phoning another, a person querying a database, a person launching a software process such as an intelligent agent search, a machine process seeking the right human consumer(s) of its information, a sensor establishing relationships with other sensors to triangulate or refine a detection, or a weapon linking to a sensor for guidance purposes.

Some characteristics of the architecture include:

- "Universal" suggests that connectivity must reach everywhere of interest. ("Of interest" is situation dependent.)
- "Ubiquitous" suggests that everything of interest must "plug in" to the connectivity. Plugging in implies some ability to interact with other plugged-in entities under some rules or circumstances—such as appropriate security.
- This "pluggability" implies standards or translators/gateways.

· Where needed access does not exist, it must be "createable" through means such as sensor deployment or establishing connectivity.

Perhaps most importantly, we need to consider "usefulness." We use the term to collectively represent a broad set of attributes that the architecture should support. First, the implementation should be user-centric and intuitive. That is, the implementations should focus on the needs and requirements of users at all operational levels of command, and support those needs in a way that minimizes reliance on specialized skills and training in the use of the architecture elements. The architecture must be adaptable and configurable. These characteristics suggest that the capabilities supported by the architecture will be totally responsive to the user's unique requirements for information to support specific missions, tasks, or functions. Finally, the architecture must be survivable in the face of all types of physical, electronic, or cyber effects, to the same degree that the user and user's physical space are survivable.

With this view, the imperatives Dynamic Interoperable Connectivity, Universal Information Access, and Focused Sensing and Data Collection apply to the architecture directly. The imperative Information Operations–Assurance and the imperative Resource Planning and Management also apply, but in the limited sense of assuring and managing connectivity and access. The Consistent Situation Representation and the Distributed Collaboration imperatives are really customers or applications that utilize the network-centric architecture rather than being fundamental elements of the architecture.

DYNAMIC INTEROPERABLE CONNECTIVITY

Dynamic Interoperable Connectivity is the conduit for all data and information, whether that information moves 15 feet or 15,000 miles. The Dynamic Interoperable Connectivity imperative aims to ensure that the warfighter has reliable and secure access to all needed information. Providing worldwide Universal Information Access requires an integrated global network for gathering and exchanging information. This includes extensive high-capacity landline connections among military users to maintain extensive databases from which warfighters may "pull." It also requires improved in-theater communications for better response to the warfighter's needs, particularly the dynamic movement of imagery and large files.

Not all connectivity users are people. Machines also must exchange data. Connectivity supporting machine data exchange has been accepted Navy practice for the four decades since the introduction of the Naval Tactical Data System and Link-11. Connectivity can involve any number of people and machines, in various locations, as required to accomplish a task. In the future, machines as users must be able to control connectivity on a priority basis.

Dynamic connectivity is flexible, supporting the time-varying needs of users. But it is also economic, supporting the sharing of resources. This allows a given set of resources to serve many times the needs that could be supported by static connections. In addition, individual users generally perform many functions and belong to multiple user communities associated with those functions. The functions may each require only part-time involvement. Connectivity requirements will then track the shifting task involvements.

The future warfighter must have full access to his/her real and virtual area of responsibility, or "operational space." The operational space may be physically small, or global, depending on the user's role. The operational space may be functionally restricted or extend beyond many organizational boundaries (for example, to include allies). Connectivity is required within and among naval nodes,[1] and between both fixed installations and mobile Navy nodes and non-Navy locations worldwide. The non-Navy locations include other Services; other U.S. government installations, facilities, and nodes; Allied forces and locations; commercial and educational entities; and even hostile forces under some circumstances. This diversity is implied by the term *interoperable*. These connectivities require a wide range of attributes. They require varying levels of security, timeliness of connection establishment, timeliness of information transfer, duration requirements for the user–user interaction, robustness against unintentional or intentional disruption, information integrity or accuracy, and simultaneity (conferencing). The varying levels for the many attributes are not set uniquely for a given connectivity—several combinations may be required for any one connection, depending on the circumstances of the moment or on diverse needs of a user performing multiple activities.

Interoperability is critical. When the community of users extends beyond Navy boundaries, interoperability based on the standards of the larger community is required. Supporting interoperability demands the ability to exchange information and commands between users. This, in turn, places demands on all of the underlying procedures, processes, and hardware at every level. Interoperability implies a common (human or machine) language, common security methods and shared "keys," common protocols, and common modulation formats or methods. Where these items are not shared in common, translation mechanisms must be provided.

Now and for the foreseeable future, the number of possible connections and the capacities of those connections between mobile or deployable nodes will fall short of total user demands. Therefore, the command organization will have to allocate available resources to users based on mission and operational needs. Some resources needed to support Dynamic Interoperable Connectivity are inherently limited. Spectrum must be shared among surveillance (both active and passive); navigation; identification, friend or foe; communications; counter-C^3; and weapons systems (soft-kill systems, in-flight missile guidance). Physical space for radios is limited, and today's radio systems (cryptographic device, modem, transmitter/receiver, antenna coupler, antenna) are usually dedicated to a single user or group. A goal for Dynamic Interoperable Connectivity at large nodes (ships, aircraft) is to eliminate dedicated equipment and spectrum. Reducing dedication of equipment and spectrum to single user classes will increase efficiency, expand the number and types of users having communications access at any given time, and reduce costs.

For very small nodes (miniature sensors, hand-held nodes), battery life is critical and energy consumption per bit delivered is a key characteristic. Universal access must be provided in a way that optimizes that characteristic.

[1] The term "node" is used to encompass manned and unmanned locations—including, for example, unmanned aerial vehicles (UAVs) and individual sensors.

UNIVERSAL INFORMATION ACCESS

A revolution in connectivity and distributed computer power is creating a potential for access to information that must be applied judiciously. Universal Information Access describes the interactive processes for information producers and information users (warfighters). The Universal Information Access imperative focuses on the warfighter's need for enough information to act appropriately, but not so much that confusion results. User pull is the "call for as needed" capability that allows the warfighter to access information, only as needed, based on changes in the operational situation. This capability requires robust information servers to support searching by forces deployed anywhere. Repositories of current, pertinent information, located at anchor desks, provide the warfighter with access to seek and receive the right information at the right time. In this paper we focus on information access by the warfighter (person), since machine information access is a subset—relying upon tools (such as intelligent agents) that could also be used by the warfighter.

The Universal Information Access imperative defines ways to meet user information needs for command and control at all levels. Warfighters must be able to access the universe of information without the need for specialized technical skills. The basic capabilities will consist of (1) user pull information transfer, (2) producer push, and (3) preplanned "information ordering."

User pull information transfer is a "call for as needed" capability allowing warfighters dynamic access to information according to mission situations. Warfighters of any rank will access the infosphere.

Producer push distributes information and alerts to customers, allowing command centers to inform and direct warfighters as needed whenever warfighters have insufficient knowledge or indications to formulate a request. Key to producer push is intelligent selection, or screening.

Preplanned information ordering has two components. First, preplanned essential information is assembled by the warfighter (at any command level) before a mission. Preplanned essential information comes from existing databases, which may be fixed in the sense that they are built and maintained independently of any specific mission. Second, information is updated as the mission requires by over-the-air updating.

User interaction is provided through (1) a *warfighter–computer interface*, (2) information assistants, and (3) information control. The warfighter-computer interface is broader in scope than a typical human–computer interface since the warfighter terminal must allow use by an automaton (an information agent) as well as by a human. The great volume of available information demands that warfighters have support in browsing, cataloging, and making sense of information—we call such support *information agents*. Such software assistants will use decision-support algorithms and artificial intelligence to help process the volume and diversity of the infosphere.

FOCUSED SENSING AND DATA COLLECTION

The developing concepts of a revolution in military affairs, or of network-centric warfare, or of operating inside an adversary's decision process, all assume availability of information upon which to base decisions and actions. Tactical decisions must be based on timely understanding, which, in turn, is based upon real-time data extracted from the area of interest.

In this imperative, *sensing* implies gathering data about the physical world through electromagnetic, acoustic/seismic, olfactory, or other measurement means. Sensing might be based on national or strategic systems including satellites and aircraft. It would include platform-based systems fielded on ships, aircraft, or unmanned vehicles. Finally, sensing might be based on deployed or dispersed tactical probes or sensor fields.

The concept of *focused* sensing implies concentration on things of interest, applying available sensing resources to obtain data and information on key subjects and areas. Focusing narrows the scope in one or more of the aspects of location, time, or type, where type refers to the events, features, or elements to be reported.

Data collection implies gathering data about the cyber world, or data about the physical world through means other than direct sensing. This would include extracting from electronic repositories, or manipulations of archived data.

The network-centric architecture extends to the sensor level. Networked sensors can collaborate to refine and enhance their data products. Some sensors will have the ability to act without real-time direction. This may involve refining their focus area, providing selective reports, or even relocating to areas of greater "interest." The primary objective is to provide the data needed by the user, who defines the focus.

INFORMATION OPERATIONS–ASSURANCE

In today's and tomorrow's world of asymmetric threats, protection of our information systems—and the network itself—is essential. Assurance in network-centric environments is less a feature of system operation than it is an empowerment of the users of these systems. Assurance features provide the access controls, authentication mechanisms, confidentiality, and integrity features that enable the users to assert their identity and to access resources in both peer–peer and client–server interactions. Assurance needs to be built into every aspect of a system in a consistent and correlated way. Piecework solutions or post-deployment appendages of assurance features are seldom successful or evolvable. The foundation of security is a clear definition of what is supposed to happen and who is supposed to perform that action. Given a clear definition of what services a system is supposed to offer and who is authorized to avail themselves of these services, assurance can be developed that these services are offered without modification, disclosure, or interruption, and that other unintended actions do not occur.

Assurance features that should be considered in the network-centric architecture include:

- Adaptation to protocol enhancement since reliance on specific protocol features can be short-lived and inflexible;
- Communication routing decisions should offer assurance of correctness. The exchange of routing information is critically important and must be communicated with assurance;
- Assurance features must support the delivery of information to multiple destinations;
- Assurance features must be designed to support joint mission execution and to support interactions with alliances of convenience;
- Interactions should be characterized as peer–peer or client–server, and

be provided. Special considerations must be made to provide services to remotely located users;

- Participants need to be identified in a consistent way throughout a system. A well-structured directory system is essential to coordinate these identifiers;
- Information should flow among people, while control flows should be contained within a site (i.e., the concept of a manager of managers is a bad idea);
- A small number of clearly defined categories of assured services should be supported. All applications that communicate must depend on one or more of these categories of services. Allowing applications to communicate in unique ways makes it very difficult to demonstrate system assurance.

Security services empower the user in the integrated interoperable distributed information sphere of the future—the network-centric architecture. The many aspects of assurance must be carefully crafted into the functional, operational, and structural aspects of information systems to serve future information warfighters.

RESOURCE PLANNING AND MANAGEMENT

Resource Planning and Management provides the tools necessary to identify and allocate resources for any given task or to meet an unplanned contingency. Such tools support effective use of limited resources including personnel, while requiring minimum manpower and skills for their use. Tools are not task-specific, and relate primarily to the planning for and allocation of C^{4}ISR electromagnetic, information processing, information management, and personnel resources. Resource Planning and Management includes:

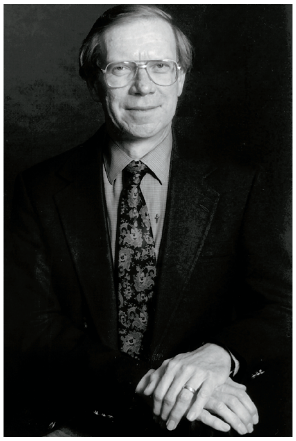
- Core services control, including self-diagnostics and healing, data storage and caching, and shared or distributed computing resources;
- The use of modeling and simulation in support of command and control;
- Decision support tools in support of focused logistics, including inventory control models, loss/damage models, and casualty models;
- Sensor tasking and collection management;
- Electromagnetic resources (antennas and other equipment; power levels; signal types and parameters; spectrum) "negotiator"—including communications resource management;
- Information management.

CONCLUSION

This paper is an attempt to identify the features of an architecture to support evolving and future network-centric operations. Recognizing these required features helps focus our energies on development of the enabling technologies to field the architecture.

AUTHORS

William L. Carper
MS in Electrical Engineering, San Diego State University, 1968
Current Research: System engineering for Naval Space Surveillance System (NSSS) Project.



Clancy Fuzak
Ph.D. in Electrical Engineering, University of Southern California, 1970
Current Research: Concepts and analyses for future naval and joint forces.

Mary Gmitruk

BS in Electrical Engineering, San Diego State University, 1985

Current Research: Roadmapping C^4ISR technologies and technology transfer.

James W. Aitkenhead

BS in Physics, San Diego State University, 1973

Current Work: Team leader for the Science and Technology Team; developing new technology for Cooperative Engagement Capability (CEC) process; and participating in the Corporate Initiatives Group (CIG).

Tom Mattoon

BS in Electrical Engineering, University of Idaho, 1970

Current Research: C^4ISR architectures and interoperability.

Victor J. Monteleon

MS in Physics and Operations Research, U.S. Naval Postgraduate School, 1966

Current Research: Chair of SSC San Diego's Corporate Initiatives Group (CIG); concepts and architectures for future Navy C^4ISR systems; C^4ISR vision development.

REFERENCE

1. Cebrowski, ADM A. K., and J. J. Garska, 1998. "Network-Centric Warfare, Its Origin and Future," *U.S. Naval Institute Proceedings*, January, pp. 28–35.

❖

Network-Centric Computing: A New Paradigm for the Military?

LCDR Lawrence J. Brachfeld, USN
SSC San Diego

ABSTRACT

This paper investigates the optimal way to implement ultra thin computer architecture into the existing Information Technology for the 21st Century (IT-21) infrastructure. Factors studied include system architecture, the effect of limited communication capabilities of naval units, changes to current battle group operating doctrine, and the benefits and risks of introducing this new capability to the Fleet.

INTRODUCTION

Network-centric warfare has been defined "as an information-superiority-enabled concept of operations that generates increased combat power by networking sensors, decision makers, and shooters to achieve shared awareness, increased speed of command, higher tempo of operations, greater lethality, increased survivability, and a degree of self synchronization. In essence, network-centric warfare translates information superiority into combat power by effectively linking knowledge entities in the battlespace."[1]

This paper discusses technology known as ultra thin clients (UTCs) and how to make information delivery more reliable and less expensive through the use of "display appliances" using a network-centric computing (NCC) architecture. The NCC approach is targeted to making information delivery simple and inexpensive. It is not a Windows-only or a UNIX-only approach, nor is it a Web browser approach that proposes to replace the inventory of existing legacy commercial off-the-shelf and government off-the-shelf applications with Web applications. The delivery of a wide variety of applications to the user is accomplished by using the network to allow servers to run applications for multiple users. Run-time environment requirements are thus confined to the servers and not propagated to all clients. Clients need only be able to accept redirected screen displays for the applications.

The main points are:

- Servers are categorized as either generic network servers or specific application hosting servers.
- Both categories of servers rely on the concept of being scaleable and taking advantage of technology to service many users.
- Clients are thin or ultra thin, relying on no application-specific code.
- Clients are not dependent on any specific operating system or hardware design.

The NCC deployment is simplified because the applications themselves are not deployed to the clients who represent the greatest number of users. For example, on a carrier with 1000 seats, there is a 1000:1 reduction in application software update costs, one application server vs. 1000 clients. Configuration management is simplified because the UTCs are zero-administration devices; all management is done at the servers. Low total ownership cost naturally follows because of the greatly reduced configuration management and network administration. High service levels are provided to the users because all applications are available over a redundant network architecture with redundant application servers that can be accessed at any UTC on the network by using smart cards.

The NCC architecture shown in Figure 1 depicts how, by using clusters of network and application servers, the display information required can be pushed out to the end user.

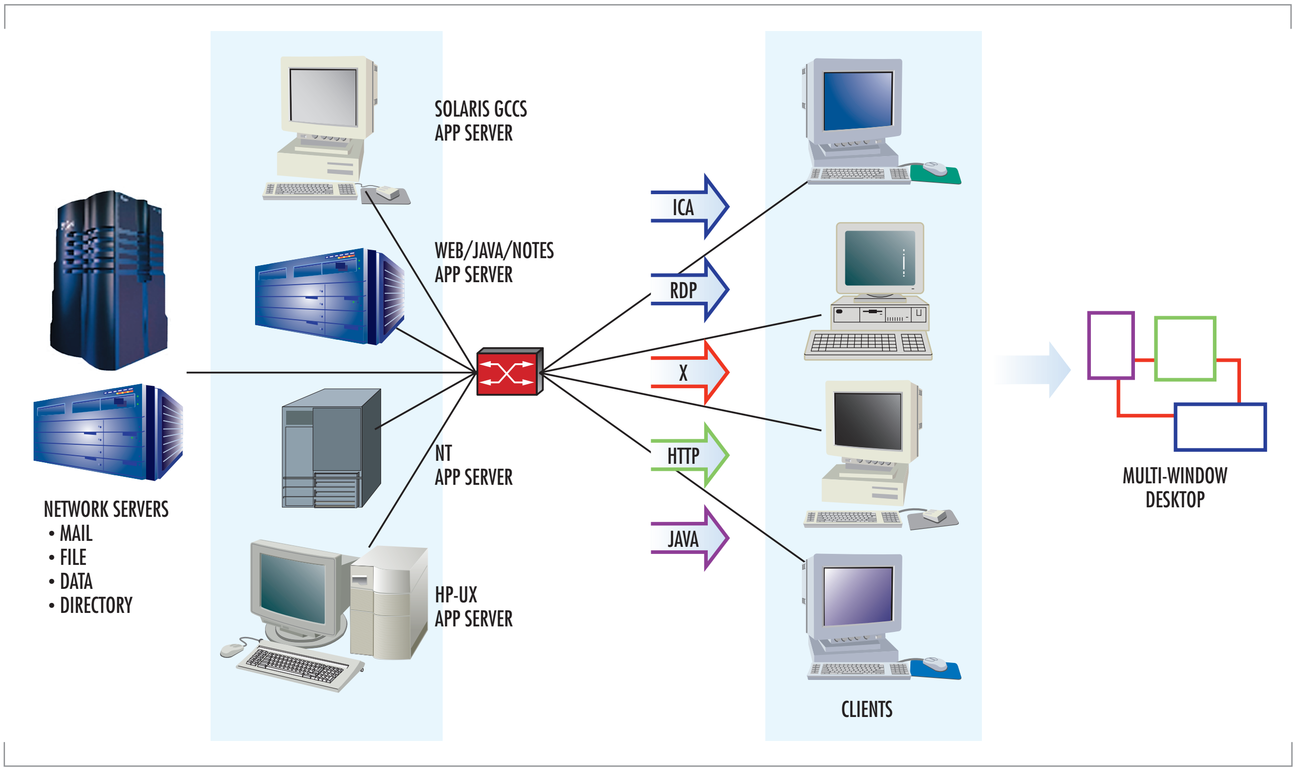


FIGURE 1. NCC architecture.

The NCC architecture has been implemented onboard USS *Coronado's* (AGF 11) Sea Based Battle Lab (SBBL) and has demonstrated the ease and flexibility with which it can be integrated into the existing Information Technology for the 21st Century (IT-21) Integrated Shipboard Network System (ISNS) local-area network (LAN). The current installation consists of 54 UTCs with seamless access to the ISNS backbone for e-mail and office automation, but it also provides access to the Global Command and Control System–Maritime (GCCS–M), GCCS–A (future capability), and Theatre Battle Management Core Systems (TBMCS) (future capability) at the users' desktops, as shown in Figure 2. Figure 3 further depicts the concept of consolidated servers. Additionally, as a natural feature of the UTC, users are no longer "tied" to their PC; they can use their smart card at any of the 54 clients and have full access to all their personal files and network applications.

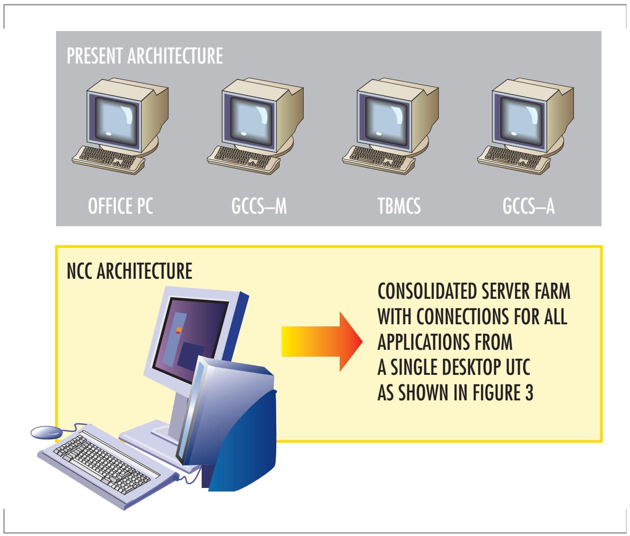


FIGURE 2. NCC desktop environment.

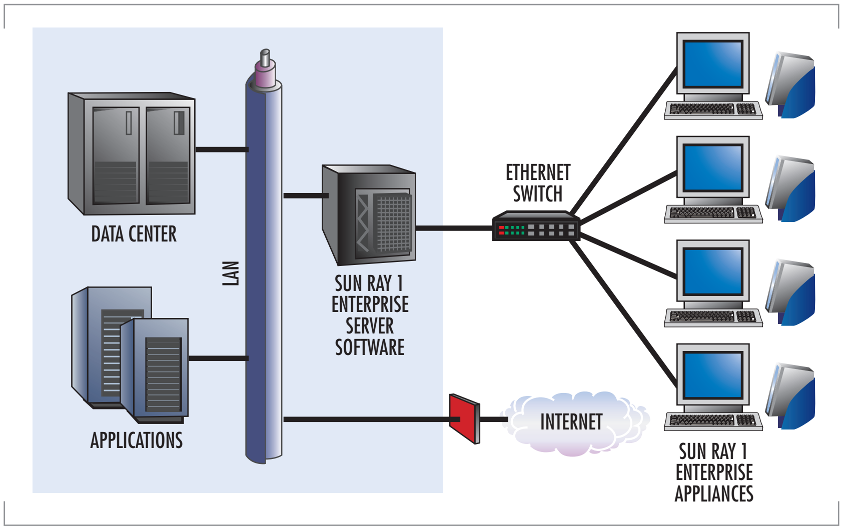

DATA CENTER
APPLICATIONS
LAN
SUN RAY 1 ENTERPRISE SERVER SOFTWARE
ETHERNET SWITCH
INTERNET
SUN RAY 1 ENTERPRISE APPLIANCES

FIGURE 3. Consolidated Server Farm (from Aberdeen Group, September 1999).

As an example, here is a brief illustration of a user's experience with the UTC architecture. Authentication happens once, when a user initially logs on the system by inserting their smart card into the slot on the UTC. There is no wait and no boot-up! The user is immediately connected to the server of their choice and has full access to the programs and files as they were left at the last logoff. The user begins working on a presentation to be given that afternoon and after a few minutes gets a call from the boss asking to see the current presentation. Prior to having a UTC, the user would have had to e-mail the draft plan or save it on a shared network drive; now, without even closing the file, the user removes his or her smart card, walks over to the boss's desk and reinserts the smart card. They are now both immediately looking at the current document, and any changes that are made are saved to the user's file either in a personal directory that only the user can access or on a shared directory to allow for additional collaboration.

As stated in Joint Vision 2020, "the overarching focus of this vision is full spectrum dominance—achieved through the interdependent application of dominant maneuver, precision engagement, focused logistics, and full dimensional protection. Attaining that goal requires the steady infusion of new technology and modernization and replacement of equipment."[2]

To meet this overarching focus with a "steady infusion of new technology and modernization and replacement of equipment" in an environment of shrinking budget resources, a radical shift from the current business model is required. The replacement and modernization of PCs to achieve this vision is impractical; thus, the UTC that never requires upgrading at the end-user location and only requires upgrades at the server level becomes the obvious choice for achieving Joint Vision 2020.

From the installation onboard the *Coronado* SBBL, it has become apparent that the users want more and more applications loaded in this architecture, which allows for the integration of legacy applications that previously required dedicated workstations or PCs. Those applications can now be accessed from any of the 54 UTCs on the network.

This architecture will mark the beginning of a new wave of computing; it is poised to redefine the distributed computing model of the networked fat client PC executing Web-based applications. Although network computing always requires computers, applications, and data, the UTC efficiently repartitions the system and redefines what goes where. By removing all computation and state information from the desktop, we truly have a zero-administration client that can help us achieve Joint Vision 2020 and reduce one of the costliest elements of information technology management.

LCDR Lawrence J. Brachfeld, USN

MS in Information Technology Management, Naval Postgraduate School, 1996

MS in Computer Science, Naval Postgraduate School, 1999

Current Research: Ultra thin client computing; multi-level security, Jini technology integration.

REFERENCES

1. Alberts, D. S., J. J. Garstka, and F. P. Stein. 2000. *Network Centric Warfare*, DoD C^{4}ISR Cooperative Research Program (CCRP), Assistant Secretary of Defense for C^{3}I, (February).
2. Chairman, Joint Chiefs of Staff. 2000. "Joint Vision 2020," (May), http:www.dtic.mil/jv2020

❖

Information Management on Future Navy Ships

Marion G. Ceruti
SSC San Diego

ABSTRACT

This paper discusses issues facing information technology (IT) system developers for Navy ships. Its overall emphasis is on the management of large volumes of tactical information that a ship must collect and process during its missions. Specifically, it describes a very ambitious notion of a future Navy Comprehensive Information Management System (CIMS). Challenges and solutions are suggested for CIMS implementation. Many technical areas of information technology are covered as a set of recommendations for future Navy information systems rather than as an analysis of problems for a particular application. Whereas they reflect the Navy's current and projected needs, many of these recommendations will be possible to achieve only with significant breakthroughs in technology and its applications. Therefore, this paper can serve as a challenge to researchers, engineers, and technology developers in government and industry to find solutions that meet future IT requirements of naval vessels.

INTRODUCTION

Because of its unique capabilities, the U.S. Navy is the primary service to achieve forward-deployed power projection as a means of protecting national interests. In the platform-centric warfare of the past, naval commanders were concerned much more with how to manage the weapons and sensors capabilities onboard their own ships and the information they could acquire than with the total tactical picture of the theater. The main reasons for this were lack of sufficient bandwidth for communications and a lack of technology to fuse, integrate, and display information rapidly. We enter the new millennium with the emphasis on information as an important resource, a condition evident in the current military trend toward network-centric warfare. Enabled by modern database management, networking, and user-interface capabilities, network-centric warfare [1] implies that all platforms in the theater are aware of and contribute to the total information available to all ships, aircraft, and ashore command centers. In some circumstances, a commander could even deploy assets based on another ship.

Network-centric warfare [1 and 2] also implies that the volume of the information available to warfighters on a theater-wide basis will keep growing. This, in turn, also necessitates that engineers provide to Navy commanders a Comprehensive Information Management System (CIMS) that includes the current capabilities of tactical and non-tactical systems. A CIMS also must feature the next-generation technologies that are now the subjects of intensive research and development efforts in the Department of Defense in general and in the Navy in particular. CIMS is not a formal Navy program; rather, it is a generic term to indicate what is expected to evolve over the next decade and beyond.

The challenge facing Navy planners and administrators is how to accomplish this in an atmosphere of cost cutting, limited budgets, and reduced resources. Whereas past military systems relied mainly on specially built equipment that conformed to military specifications, today's Navy and that of tomorrow will feature a greater usage of commercial off-the-shelf hardware and software. This trend will enable not only cost savings but also the use of new products and services of industry to maintain the leading edge in technology for the warfighter. (See, for example, [3]).

Also consistent with the policy of cost savings, next-generation Navy ships will have fewer personnel. Sailors will need to learn multiple jobs and become familiar with multiple tasks, in addition to what they are doing today. This implies that more automation will be necessary in all areas, especially automation in training systems, such as Web-based training. More than ever, tomorrow's Navy will learn how to accomplish more with fewer resources.

CIMS CAPABILITIES

The ideal CIMS will provide the following capabilities and address the following topics:

- Database integration and knowledge-base integration
- Knowledge-base integration with databases
- Database and knowledge-base standards and refresh for these standards
- Maintenance of security during database integration
- Data standardization to facilitate database and knowledge-base integration
- Data warehouse technology and data warehouse software refresh
- Data preprocessing and cleansing prior to storage in data warehouse
- Data mining that includes mission-directed Web searches
- Data-mining tool refresh
- Enhanced data-fusion technology
- Advanced data storage systems
- User-friendly database and knowledge-base access
- Database and knowledge-base management, including correct database management system (DBMS) and knowledge-base management system selection and refresh of commercial off-the- shelf and government off-the-shelf and software
- Regular updates of standard command and control systems, such as the Global Command and Control System–Maritime (GCCS–M) [4] and the databases that support them
- Periodic assessment of data storage requirements and plan to meet future needs
- Use of intelligent agents in conjunction with data warehouse, databases, and knowledge bases
- Knowledge- and data-replication to avoid a single point of failure
- Subsystem to provide situational awareness
- Computer network information on all offensive efforts
- Information-service "reach-back" to networked ashore capability
- Information warfare activity integration
- Integration of intelligence and security information

CHALLENGES AND SOLUTIONS FOR CIMS IMPLEMENTATION

The Navy must overcome many obstacles before the completion of a CIMS. This section describes some of these obstacles and challenges [4] that Navy information systems engineers will encounter.

Data Fusion

Naval forces need to link and fuse in real time more sensor data from a wide variety of sources. This implies a requirement for a modular, open-systems environment in which various data fusion engines can be inserted or deleted. Meeting this requirement necessitates an unprecedented data fusion effort for sensors on aircraft, unmanned airborne vehicles, satellites, and precision weapons of all U.S. and allied forces. The Navy will fuse information, or will used finished fused data products, from other services and allies in the common operating picture. The CIMS ideally will accommodate any sensor input—a situation that is very open-ended. Therefore, one challenge is for the U.S. Navy to know when this requirement is satisfied, especially when the Navy has no direct control over the interface designs of sensors from the other services and allies. (For more information on the joint vision, see [5 and 6]).

Distributed Database Components

Data will be collected and integrated. For example, the CIMS will contain the biological and chemical sensor information that will be integrated. Engineers will need to develop metadata [7] documentation of database systems components with an explanation of the relationship between components (e.g., how their data elements are subsets or a superset, etc., of the integrated databases) that support major existing systems, such as GCCS–M. Database access efficiency depends on the hardware, the DBMS, the operating system, and the relative priorities of competing tasks. Thus, the CIMS will feature a modernized version of a distributed, federated database. (See, for example, [1, 8, 9, and 10]).

Optimized Data Structure

The establishment of an information warehouse in a data management system for all users is not enough to guarantee an optimized data structure. Therefore, engineers must consider all of the factors necessary to achieve an optimized data structure. Also, engineers must provide to the users (e.g., sailors) an online document that will explain the operations for which the data structure will be optimized. For example, a data structure optimized for retrieval performance will not be optimized for data storage performance and vice versa [11]. The documentation will list the advantages and disadvantages of the particular data structure selected for implementation. This information is generally not present in current command and control database systems in any comprehensive sense.

Data and Database System Standardization

The CIMS will feature data standardization that is needed, not only for sensor-data fusion, but also for other aspects of data integration. The CIMS will contain an up-to-date reference list of all necessary and germane data standardization documents. The Defense Information Systems Agency (DISA) has instituted the Defense Information Infrastructure Common Operating Environment (DII COE) as an essential element for inter-service interoperability [4]. The DII COE includes the Shared Data Environment. The CIMS will comply with DISA's standards at each level of DII COE.

Data Aggregation

The CIMS will provide access to distributed legacy databases through a user interface, which is a step toward data aggregation. However, this is insufficient to guarantee uniform data services to all active components. It is only a step on the way to data integration and not data integration in its entirety. The challenge that the Navy faces here is to determine all steps in the information integration process, including data aggregation and addressing any security implications that this aggregation creates [4] on a resources-available basis.

Information Integration

Information Integration Analysis

Extensive analysis is necessary to integrate and present clear and non-redundant information. The Navy will face the challenge of ensuring that the CIMS will be based on the analyses that have been performed, considering the cost and security implications. The ideal CIMS will use what engineers have learned from others' experiences in information systems integration so it can present clear, useful, timely, and non-redundant information to its users.

Information System Integration Details

The Navy will need to describe and document its integration approach, including how much integration can be completed given the financial constraints. To accomplish a successful CIMS, engineers will need to provide details of how information systems integration will be performed on all levels, including semantic and data levels of integration. The engineers will become familiar with the integration methodology and the algorithms used to accomplish it. A list of integration priorities must be developed because all desired integration tasks cannot be performed in a reasonable timeframe and within budget [12].

Online documentation will describe the database integration strategy and or methodology with enough detail so that personnel who are not computer experts will know that the integration method will result in the required seamless database interfaces and will include integration on all levels. Data residing at different decision centers will not be consistent automatically. Therefore, the CIMS will need to be able to identify and resolve the inconsistencies. (See, for example, [9 and 13]). The integration method and architecture will be specified. The level of integration in the CIMS will be specified so that the user will know what the developers could accomplish at the allocated funding level. Ideally, the CIMS should be integrated on three levels: platform, syntactic (data model), and semantic [9 and 13].

Integration Methods and Large-screen Displays

Large-screen displays are a common feature of modern command centers. Large-screen displays can facilitate error detection on an *ad hoc* basis, but they cannot substitute for a thorough database integration effort. To reduce inconsistencies in the data, more automated methods are needed and specific algorithms should be utilized. The CIMS should provide a description of all integration methods that will be used before giving users a possible means (but not a systematic method) to notice data inconsistencies via large-screen displays.

Data Cleansing

The ability of an information warehouse, a common backbone, and a large-screen display to increase reliability and consistency is only as good as the integration and data cleansing [14] that has taken place in the data sources. This integration and data cleansing must be performed before taking the following steps:

- Installing the data in the warehouse
- Making data available on a common backbone
- Displaying them on large-screen displays

Ideally, only clean and consistent data will be stored in the information warehouse. However, few if any databases of appreciable size have ever had totally clean and consistent data.

Information Warehouse

In the ideal CIMS, an information warehouse provides a complete source of warfighting information and knowledge to all echelons. To accomplish this, engineers will have to define metrics to evaluate the completeness of warfighting information and knowledge in the information warehouse. They will need to test and evaluate the ability of the information warehouse [15] to deliver information efficiently to the user at each stage of compliance. For example, it may be possible to provide a 70% solution at time, t, and an 80% solution at time, t+x. The CIMS will function best if database administrators load all warfighting information and knowledge into the information warehouse in well-defined stages. A difficult challenge to engineers will be to determine how all data systems will be integrated into a single information warehouse. A common metadata repository must be part of the data warehouse to support the CIMS and the common operating picture.

Knowledge Management

Knowledge Standards and Knowledge Management

Commercial, open-system standards will contribute to an affordable and information system architecture designed to accept upgrades efficiently. Database management services with Relational Database Management Systems and with Object Relational Database Management Systems, such as Open Database Connectivity, are well known. However, standards as they apply to existing capabilities and equipment are insufficient. Because knowledge centricity is an important feature of future ships, the information processing standards will need to include the emerging knowledge standards, such as Open Knowledge-Base Connectivity (OKBC) [16], Knowledge Interchange Format, or Knowledge Query Markup Language [17]. OKBC is the knowledge analog of Open Database Connectivity. Standards need to support open data and information exchange architecture. The CIMS will support this criterion by including the class of standards to address knowledge interchange. The current knowledge standards will evolve to higher levels during the coming decade. Therefore, the CIMS will be evaluated for a periodic refresh of knowledge standards as new ones emerge. These standards will contribute to database and knowledge-base integration, including the integration of ontologies necessary to support future artificial-intelligence technology in the knowledge management system(s) of the CIMS.

Common Ontology

The CIMS will have a common ontology and a knowledge base derived from it that will be accessible to all users over the network. This ontology will be necessary to enable the semantic integration that knowledge centricity implies [17]. In addition to OKBC, a common ontology and the tools to merge ontologies and knowledge bases (of other services and allies) are necessary pieces of the puzzle [16 and 17]. The ideal CIMS will include the complete integration of knowledge bases and databases into a seamless common operating picture. The Defense Advanced Research Projects Agency has sponsored the High-Performance Knowledge Base program, which produced results that can contribute directly to information-systems and ontology integration problems. (See, for example, [16, 17, 18, and 19]). The CIMS also will make a common ontology available to intelligent software agents. The Navy's challenge in this area is to identify the correct ontologies for integration and to include all relevant concepts in the unified ontology.

Data Mining

Data fusion processing and planning processing are necessary but insufficient by themselves to ensure functional knowledge-centric decision centers. Tactical data mining will be a capability exploited on future Navy ships. The CIMS will assist users to perform the steps of data mining to be carried out on each ship. The CIMS also will assist users in determining the desired outcomes of data mining for a particular task and the tactical data-mining tools required to complete the task. The CIMS will integrate the outputs of the intelligent software agents and coordinate the behavior of the intelligent software agents with each other with the output of the tactical data-mining tools to augment the knowledge base. Promising current approaches to data-mining problems [20, 21, 22, 23, 24, 25, and 26] in the area of command and control [20, 21, 24, and 25] range from Bayesian networks for data-classification tasks [21] to knowledge mining with randomization and features to overcome the knowledge-acquisition bottleneck [25].

Mission-directed Data Mining

Although Internet connectivity is common in today's Navy and will be part of the total communications package, a specific need for this type of connectivity has been identified to support cryptologic and information-operations, mission-directed Web searches. The CIMS will enable sailors to implement cryptologic and information-operations, mission-directed Web searches, and to integrate the information obtained from such searches with other data sets already in the database where appropriate [1].

Data-mining Technology Upgrades

Tactical data mining is not a reality today. The whole data-mining process as we know it typically takes too long to be accomplished in seconds and is therefore not yet suitable for tactical, real-time applications. However, in the coming decade, tactical data mining may be not only possible, but tools to accomplish it may be modular, commercial off-the-shelf, user-friendly, and compliant with Department of Defense standards. Therefore, the CIMS will include technology refresh in the area of data mining to enable this new and developing technology to contribute significantly to knowledge centricity.

Information Operations, Efficiency, and Security

Information Operations

The information management and information integration activities will be coordinated with the information operations activities to provide efficient and seamless information services. An ideal CIMS will be able to handle smooth interoperation and conflict resolution between these activities.

Situational Awareness

Situational awareness relates to the common operating picture, the common tactical picture, etc., that will be available to all Navy personnel in the theater. Tactical decision-makers on future ships will have an adequate situational awareness about their operational posture (friendly, hostile, and neutral) in the electromagnetic spectrum, in the computer network environment, and in other domains such as the psychological, cultural, and environmental "pictures." Inherent in this requirement is the need for appropriate decision aids, algorithms, displays, simulation tools, etc., to provide situational awareness in the information-operations arena.

Computer Network Exploit and Attack

A Strike Force Commander must be aware of all offensive efforts that may affect the strike (hard kill or soft kill). The future CIMS will need a specific capability to provide the Strike Force Commander (and other Strike Force Commanders in the theater) information on all offensive efforts to avoid overkill of targets that could cause the unnecessary expenditure of scarce and/or limited ordnance resources.

Reach-back

To reduce the most expensive cost factor (payroll), personnel limits have been specified for future ships, with the expectation that functions can be moved ashore and future ship operators can "reach back" for what they need. To make sure that future ship personnel will have all the information services they need at the same level of reliability, these supporting shore services will need to be more secure, robust, redundant, and capable than they are today. An ideal CIMS will need to meet all of the information system requirements, either onboard the ship, in the theater, or on shore.

Information Warfare Activity Integration

The Navy divides information warfare into two categories: (1) offensive (information attack) and (2) defensive efforts (information protection and assurance). The ideal CIMS will integrate these functions aboard future ships, for both traditional information systems, e.g., tactical communications, message traffic, voice, etc., and those associated with the computer network environment.

Levels of Security

Secure database technology is now available. The CIMS will feature multi-level security (MLS), which will address issues such as MLS vs. network security, network security vs. secure operating system and/or secure DBMS, etc. Security needs to be implemented at all levels to preclude a weak link in the security chain. Network security is not enough. Most of security is enforced on the network in a network-centric security system. The CIMS will provide security at the operating systems and database management systems level.

SUMMARY

The information management systems on future Navy ships will provide rapid access to fused and integrated data and knowledge to meet the ever growing needs of tomorrow's warfighter at sea. The technology now in the research and development stages will make a valuable contribution to enhance the capabilities of our naval and joint forces throughout the coming decades.

ACKNOWLEDGMENT

The author thanks the Naval Sea Systems Command for financial support of this effort.

Note: This paper will also appear in the *Proceedings of the AFCEA 18th Annual Federal Database Colloquium and Exposition* on "Database Scalability, Quality, and Interoperability," 28–30 August, 2001, San Diego, CA.

Marion G. Ceruti

Ph.D. in Chemistry, University of California at Los Angeles, 1979

Current Research: Information systems analysis, including database and knowledge-base systems, artificial intelligence, data mining, cognitive reasoning, software scheduling and real-time systems; chemistry; acoustics.

REFERENCES

1. Ceruti, M. G. 1999. "Web-to-Information-Base Access Solutions," *Handbook of Local Area Networks*, chapter 4-5, pp. 485–499, Auerbach Publications, Boca Raton, FL.
2. Ceruti, M. G. 2001. "Mobile Agents in Network-Centric Warfare," *Proceedings of the 5th IEEE International Symposium on Autonomous Decentralized Systems* (IEEE ISADS 2001).
3. Johnson, J. L., CNO, U.S. Navy. 1997. "Forward . . . from the Sea: The Navy Operational Concept," http://www.chinfo.navy.mil/navpalib/policy/fromsea/ffseanoc.html
4. Ceruti, M. G. 1998. "Challenges in Data Management for the United States Department of Defense (DoD) Command, Control, Communications, Computers and Intelligence (C^4I) Systems," *Proceedings of the 22nd Annual International Computer Software and Applications Conference, IEEE COMPSAC'98*, pp. 622–629. Also, see http://www.disa.mil/disahomejs.html
5. Defense Technical Information Center, Joint Vision 2010, http://www.dtic.mil/jv2010/
6. Defense Technical Information Center, Joint Vision 2020, http://www.dtic.mil/jv2020/jvpub2.htm
7. Foss, R. and R. Haleen. 1997. "An Environment for Metadata Engineering," *Proceedings of the 14th AFCEA DoD Database Colloquium '97*, pp. 75–91.
8. Ceruti, M. G. and S. A. Gessay. 1998. "White Paper on the Next-Generation Data-Access Architecture for Naval C^4I Systems," *Proceedings of the 15th AFCEA Federal Database Colloquium '98*, pp. 451–472.
9. Ceruti, M. G. and M. N. Kamel. 1994. "Semantic Heterogeneity in Database and Data Dictionary Integration for Command and Control Systems," *Proceedings of the 11th AFCEA DoD Database Colloquium '94*, pp. 65–89.
10. Putman, J., B. M. Thuraisingham, W. Chitwood, and M. G. Ceruti. 2000. "Experience in Developing an Information-Sharing Environment in a Large Government Enterprise Using WWW, Federation, Business Components, and Data Warehousing Technologies," *Proceedings of the 17th AFCEA Federal Database Colloquium and Exposition*, pp. 229–244.
11. Ceruti, M. G. 1996. "Development Options for the Joint Maritime Command Information System (JMCIS) Specialized Data Servers," *Proceedings of the 13th AFCEA DoD Database Colloquium '96*, pp. 217–227.
12. Ceruti, M. G., B. M. Thuraisingham, and M. N. Kamel. 2000. "Restricting Search Domains to Refine Data Analysis in Semantic-Conflict Identification," *Proceedings of the 17th AFCEA Federal Database Colloquium and Exposition*, pp. 211–218.
13. Ceruti, M. G. and M. N. Kamel. 1998. "Heuristics-based Algorithm for Identifying and Resolving Semantic Heterogeneity in Command and Control Federated Databases," *Proceedings of the IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'98)*, pp. 17–26.
14. Ceruti, M. G. and M. N. Kamel. 1999. "Preprocessing and Integration of Data from Multiple Sources for Knowledge Discovery," *International Journal on Artificial Intelligence Tools (IJAIT)*, vol. 8, no. 2 (June), pp. 152–177.
15. Malloy, K. 1998. "Data Warehouse Requirements: Scalability, Availability and Manageability," *Proceedings of the 15th AFCEA Federal Database Colloquium '98*, pp. 569–577.
16. Chaudhri, V. K., A. Farquhar, R. Fikes, P. D. Park, and J. P. Rice. 1998. "OKBC: A Programmatic Foundation for Knowledge Base Interoperability," *Proceedings of the 15th National Conference on Artificial Intelligence*. (Also as KSL Technical Report KSL-98-08).

17. Ceruti, M. G. 1997 "Application of Knowledge-base Technology for Problem Solving in Information-Systems Integration," *Proceedings of the 14th AFCEA DoD Database Colloquium '97*, pp. 215–234.
18. Fikes, R., A. Farquhar, and J. P. Rice. 1997. "Tools for Assembling Modular Ontologies in Ontolingua," *Proceedings of the Fourteenth National Conference on Artificial Intelligence.* (Also as KSL Technical Report KSL-97-03).
19. Lin, A. D. and B. H. Starr. 1998. "HIKE (HPKB Integrated Knowledge Environment)–An Integrated Knowledge Environment for HPKB (High Performance Knowledge Bases)," *Proceedings of the IEEE Knowledge and Data Engineering Exchange Workshop, KDEX'98*, pp. 35–41.
20. Ceruti, M. G. 2000. "The Relationship Between Artificial Intelligence and Data Mining: Application to Future Military Information Systems," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, p. 1875.
21. Ceruti, M. G. and S. J. McCarthy. 2000. "Establishing a Data-Mining Environment for Wartime Event Prediction with an Object-Oriented Command and Control Database," *Proceedings of the Third IEEE International Symposium on Object-oriented Real-time Distributed Computing, ISORC2K*, pp. 174–179.
22. Ganti, V., J. Gehrke, and R. Ramakrishnan. 1999. "Mining Very Large Databases," *IEEE Computer*, vol. 32, no. 8 (August), pp. 38–45.
23. Han, J., L. V. S. Lakshmanan, and R. T. Ng. 1999. "Constraint-Based Multidimensional Data Mining," *IEEE Computer*, vol. 32, no. 8 (August), pp. 46–50.
24. Ramakrishnan, N. and A. Y. Grama. 1999. "Data Mining: From Serendipity to Science," *IEEE Computer*, vol. 32, no. 8 (August), pp. 34–37.
25. Rubin, S. H., M. G. Ceruti, and R. J. Rush, Jr. 2000. "Knowledge Mining for Decision Support in Command and Control Systems," *Proceedings of the 17th AFCEA Federal Database Colloquium and Exposition*, pp. 127–133.
26. Thuraisingham, B. M. and M. G. Ceruti. 2000. "Understanding Data Mining and Applying it to Command, Control, Communications and Intelligence Environments," *Proceedings of the 24th IEEE Computer Society International Computer Software and Applications Conference, COMPSAC 2000*, pp. 171–175.

❖

Object Model-Driven Code Generation for the Enterprise

William J. Ray
SSC San Diego

Andy Farrar
Science Applications International Corporation (SAIC)

ABSTRACT

This paper discusses the benefits of using a code generator to synthesize distributed, object-oriented servers for the enterprise from object models. The primary benefit of any code generator is to reduce the amount of repetitive code that must be produced, thus saving time in the development cycle. Another benefit to our approach is the ability to extend the services generated, enabling the code generator to act as a force multiplier for advanced programmers. Having a code generator synthesize complex code dealing with concurrency, replication, security, availability, persistence, and other services for each object server will ensure that all servers follow the same enterprise rules. Also, by using a code generator, developers can experiment more easily with different architectures. One of the final benefits discussed in this paper is that when using a code generator for the data layer of enterprise architecture, changes in software and evolving technology can be handled more readily.

INTRODUCTION

Joint Task Force–Advanced Technical Demonstration (JTF–ATD) was a Defense Advanced Research Projects Agency (DARPA) project in the field of distributed, collaborative computing. In a typical JTF command hierarchy, the critical people, relevant data, and their supporting computers are geographically distributed across a wide-area network. This causes many problems that would not exist if they were all in the same location. The goal of JTF–ATD was to make it easier for people to work together. A system that facilitated the sharing of data and ideas without compromising security, timeliness, flexibility, availability, or other desirable qualities was needed. After experimentation with numerous architectures and implementations, the JTF–ATD concluded that an enterprise solution to data dissemination and access was needed. It also became apparent that the different types of data needed to support JTF missions were as ubiquitous as the missions themselves. Therefore, planning systems would need the ability to associate previously unknown data elements to their plan composition. A distributed, object-oriented design held the most promise to meet these goals.

Unfortunately, building distributed, object-oriented data servers with the complex infrastructure to support enterprise solutions was costly and time consuming. JTF–ATD built the Next Generation Information Infrastructure (NGII) toolkit to address this problem. The NGII toolkit allows developers to code generate object-oriented data servers in days rather than months. The NGII code generator synthesized complex code dealing with concurrency, replication, security, availability, and persistence for each server, thus ensuring that all servers followed the same enterprise rules. The NGII toolkit and its descendant, Quava, are widely used by many projects today to help generate distributed, object-oriented servers with the intelligence to act in concert across the enterprise. Quava is available to the public and can be downloaded at
http://www.saic.com/quava/.

RELATED WORK

Work related to the topics discussed in this paper includes research in program synthesis, code generation, software prototyping, software reuse, software engineering, and software maintenance.

Although much of the research in the fields of program synthesis and code generation deals mainly with optimization, the process of generating code for optimizing digital signal processors (DSPs) or machine language has many similarities to the generation of code for an enterprise data layer. In earlier work, several researchers have generated code from descriptive languages or object models [1, 2, 3, and 4]. Whether the code generated was machine language or code that needed to be compiled is not material to the process of generating the code from a more abstract foundation.

Some researchers even took the generation of code a step further to aid in the creation of control code for multiple processes. In the Computer-Aided Prototyping System (CAPS), code is generated from a more abstract language to simulate a real-time system [5 and 6]. Attie and Emerson synthesized concurrent programs from temporal logic specifications [7].

Software reuse has always fallen short of its lofty goals. The reasons cited for its failure are too numerous to list [8]. Some of the most promising work to help reach the goals of software reuse involves a hybrid approach of program synthesis by making use of reusable code components and code generation [9]. This approach is the one taken by the tools described in this paper.

CODE GENERATOR

Quava provides application developers with an Integrated Generation Environment (IGE) that allows them to convert engineering designs from Computer-Aided Software Engineering (CASE) tools (e.g., Rational Rose, Oracle Designer, etc.) into Unified Modeling Language (UML) encoded design objects. Quava can then generate implementation code that can incorporate Common Object Request Broker Architecture (CORBA), Remote Method Invocation (RMI), Component Object Model (COM), or Java 2 Enterprise Edition (J2EE) services. The developer has complete control over which services, architecture, and language to use for their application.

Design

The Quava system is composed of four basic pieces (Figure 1).

The first piece, the repository adapter, imports data and can communicate with commercial off-the-shelf (COTS) modeling tools, such as Rational Rose or Designer 2000, or read models stored in the Object Management Group's (OMG's) XML Metadata Interchange (XMI) file format. XMI is key to interoperability with other COTS modeling tools. The repository adapter imports a model, which is then instantiated as a UML 1.3 metamodel. Internally, Quava can store its UML

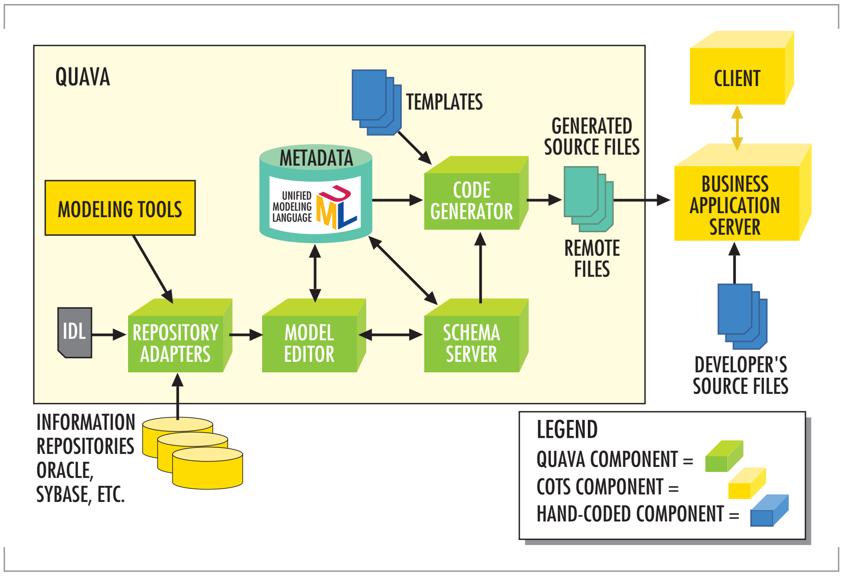


FIGURE 1. Code generation system.

objects either in an XMI flat file or to a UML server, called the Schema Server, for enterprise-wide sharing of models.

The second piece is a tool for altering the UML model. While Quava is not a modeling environment, we did allow for model editing because many COTS tools only support older versions of the UML standard, and many do not support the kinds of additional modeling information designers may want to express. Quava provides the Model Editor, which allows a user to go in and change or add information to the UML model. One example of this is the mapping of one model to another. This is very common when mapping from an application object model to a database model. Model-to-model mapping can also occur between different UML models to automatically generate interface code from a specific model to a shared model. In creating any additional modeling information, Quava still maintains the UML standard by only using UML metamodel objects to represent the additional information. This allows changed models to remain compatible with other COTS modeling tools.

The third piece is a set or sets of templates that guide and direct the generator to precisely what code to produce. Quava differs from many code generators that are used to produce code for a specific COTS tool or environment. Quava users can change or add new templates to allow production of any type of output in any language. The templates are written in either ECMAScript, which is a standardized version of JavaScript, or in Java. The templates allow for maximum flexibility and provide a mechanism for the users to define both the code output and the process flow the generator takes through the model.

The fourth piece is the generation engine. The generation engine pulls in a UML model and then proceeds to apply the selected set of templates against the different elements of the UML model. Processing continues until all selected templates have been processed against the model. Finally, unlike many other tools, the code produced is not tied to Quava in any way and can be imported into whatever development environment the user typically uses.

TEMPLATES

The templates that drive code generation are the key to both the generation's output and the level of control the user has over the generation process. During the course of experimentation with the model-driven code generation approach, we focused on three main issues. The first issue was identifying which types of services lend themselves to model-based code generation. The second issue was how code generation could help with the composition of services in a large-scale architecture, and the third issue was how easily the templates could be extended or new templates added.

Types of Services

To identify which services best lend support to a model-based code generation approach, we focused on where developers spend most of their time. Current software products allow users to generate skeleton code for different architectures, but this code is limited to just a single architecture and does not help with any of the actual logic of the objects. So, where would users get the most "bang for the buck"? Architectural services. Architectural services came to the forefront because they require the developer to implement additional functionality into each object in the schema in support of the service. For example, an Extensible Markup Language (XML) streaming service may provide a class library for creating the stream and sending and receiving a stream, but the objects within the system will need to implement a method to serialize their attributes to an XML stream. This type of service, where knowledge of the model can reduce the amount of work a developer has to do, is exactly where the code generation process fits in. Below is a very simple ECMAScript template for generating a method to serialize an object to an XML stream.

```
/****************************************************************/
// Xml Example/
function writePackage(modelhdl)
{
 var i, interfaceName;
 // Get All the element in this model
 classesList = modelhdl.getOwnedElement();
 // Loop through each element in the model
 for(i = 0; i < classesList.size(); i++) {
   // GLOBAL class object
   xmlClassObj = classesList.elementAt(i);
   // If it's a class then process it otherwise look for nested packages
if(xmlClassObj.getClass().getName() =="mil.darpa.ngii.uml.umlClass")
      writeClass(xmlClassObj);
      else                              if(xmlClassObj.getClass().getName()==
"mil.darpa.ngii.uml.Package")
      writePackage(xmlClassObj);
    }

   }


   /***************************************************************
******************/
   /**
    * Write the class structure: header, attributes, and footer.
    */
   function writeClass(xmlClassObj)
   {

    myXMLFile.writeln("public void writeToXML(StringWriter out)");
    myXMLFile.writeln("{");
    myXMLFile.writeln(" out.write(/"<class>/");");
    myXMLFile.writeln("
out.write(/"<classname>"+xmlClassObj.getName()"+</classname>\n/");
");
   myXMLFile.writeln(" out.write(/"<attributes>\n/");");
     writeAttributes();
   myXMLFile.writeln(" out.write(/"</attributes>\n/");");
   myXMLFile.writeln(" out.write(/"</class>/");");
   myXMLFile.writeln("};");

   }
```

```
    /*************************************************************
******************/
  /**
   * Write-out attributes, operations, associations, etc. of a class.
   */
   function writeAttributes(xmlClassObj)
   {
    featureVector = xmlClassObj.getFeatureList(null);

   for (i=0;i<featureVector.size();i++)
   {
     thisFeature = featureVector.elementAt(i);
     thisFeatureType = new
java.lang.String(thisFeature.getClass().getName());
     if (thisFeatureType.equals("mil.darpa.ngii.uml.Attribute"))
     {
     myXMLFile.writeln("out.write(/"<attribute>\n/");");

myXMLFile.writeln("out.write(/"<name>"+thisFeature.getName()+"
</name>\n/");");

myXMLFile.writeln("out.write(/"<type>"+thisFeature.getType().
getName()+"</type>\n/");");

myXMLFile.writeln("out.write(/"<value>/"+"+thisFeature.getName()+"
+/"</value>\n/");");
       myXMLFile.writeln("out.write(/"</attribute>/");");
      }
     }
   }
```

This portion of template code when applied to a simple class:
Class A with attributes:

String name
String address
long age

would produce the following code:

```
public void writeToXML(StringWriter out)
{
 out.write("<class>");
 out.write("<classname>A</classname>");
 out.write("<attributes>");
out.write("<attribute>");
out.write("<name>name</name>");
out.write("<type>String</ type >");
out.write("<value>"+name+"</ value >");
out.write("</attribute>");
out.write("<attribute>");
out.write("<name>address</name>");
out.write("<type>String</ type >");
out.write("<value>"+ address +"</ value >");
out.write("</attribute>");
out.write("<attribute>");
out.write("<name>age</name>");
out.write("<type>long</ type >");
out.write("<value>"+age+"</ value >");
out.write("</attribute>");
 out.write("</attributes >");
out.write("</class>");
};
```

Service Composition

Service composition is the second area we focused on, and it proved to be the most challenging. Composing components within a system is usually a process of plugging in interfaces to well-defined units of functionality, such as Java Beans. Composition of services within an object in a systems schema is much more difficult. We discovered and implemented a number of different ways to compose services without affecting other aspects of the objects although each comes with its own unique issues. The first approach we took was to have the template developer insert calls to outside functions/methods at the correct place in the generation process. This approach, while it did work, did not prove to be very scaleable to a large number of different services because of the knowledge required about each service by the template developer. The second approach was to allow a template developer to implement a set of interfaces, which get calls based on the type of interface or based on template execution. This approach proved to be much more scaleable to a wide number of optional services, but does require the template developer to be much more versed in software development because it currently works only with the Java templates.

Template Modification and Addition

Our third area of focus was the ease of extending and adding new templates. Templates can currently be written in either Java or ECMAScript. Java templates allow for many developers to use the same language that they are using to code their templates. ECMAScript allows developers who have used VBScript or JavaScript to jump in and begin making use of a powerful development tool.

Our conclusion from our work with the code generation template was to concentrate on the Java-based templates. This conclusion was reached based on having the power of a full object-oriented programming language and using the language most developers were familiar with. In addition, because experts in the different areas of software development are usually the people writing templates, they prefer to write in a language that they commonly use.

BENEFITS

Many of the benefits of code generation are obvious, such as the decrease in time to market of new applications and systems, reduction in the amount of new code to be tested, and a reduction in the number of human errors. In this section, we will explore time reduction and some of the other benefits of code generation.

Code generation allows reuse of one of the scarcest resources in most companies: specialized experts. Experts in distributed transactions, security, or concurrence can be used to write specialized templates, thus allowing for corporate capture of that specialized knowledge and providing a force multiplier to other developers in an organization. Code generation also allows groups to define how they want the code to "look." Styles and enterprise-wide coding standards can be enforced by using templates that follow the standards. Because Quava allows the user to select which sets of templates to apply to their model, developers can experiment with a wide array of architectures and design patterns to see which best fits their specific requirements. Finally, code generation allows developers to be free of their underlying technology. Currently, when a new technology comes out, the developer must go back and re-code an application or system to make use of it. With code generation, new technologies can be merged with current systems, or underlying technologies can be completely replaced by new technology.

Reduction in Development Time

A reduction in development time is the main reason for using code generation techniques. Quava allows the developer to jump straight from the design into the coding phase with very little effort. Normally, the developer is handed a design document and must start from, at best case, generated code skeleton, or at worst case, from scratch. Quava reduces the amount of code a developer must write far more than generators that provide a code skeleton because it is generating object behavior, not just code file structure. Take the example used above for an XML streaming method. This would not be hard to write by hand, but why waste the developer's time doing something that could be generated? A reoccurring benefit of generating methods such as the XML streaming is that any time the model changes, those changes are quickly reflected in the source code. Eliminating human errors that result from typos and simple logic errors also reduces development time. Once a template has been tested, the code that it produces requires far less code testing, allowing the tester to focus more on the business logic of the system.

In our research on code generation, we measured a number of projects with varying object schemas to gather some quantifiable numbers of the kinds of savings code generation could produce. Table 1 shows values captured from some of these projects. The values for lines of code generated have been rounded off to the nearest thousand.

Overall, code generation has been proven to increase the speed at which systems and applications can be implemented, and, with Quava's generation technologies, the reduction is magnified by the experience of the developer.

TABLE 1. Code generation case study.

Case	Number of Classes	Average Attributes per Class	Average Operations per Class	Lines of Code Generated
A	7	30	24	8,000
B	120	22	6	257,000
C	321	12	8	750,000

Force Multiplier

Concurrency issues, complex services issues, and other difficult programming tasks can be encapsulated in templates. By having your best software engineers develop templates, every software engineer that generates an object server with that template may take advantage of their knowledge. In essence, with a software development model where experts create templates and junior programmers develop applications using object servers generated from such templates, an organization can produce much more high-end software. Of course, the exact value to the organization is only measurable by the number of times a template can be used.

Standardization of Enterprise Rules

By code generating the entire set of object servers with the same templates, a system engineer is guaranteed adherence to these enterprise rules. Different developers can interpret enterprise rules differently. Ambiguities in the software requirements specification can lead to major additional costs later on in the software development process [10].

If developers are allowed to produce object servers with different tools or different templates, it is impossible to guarantee that the system will perform as intended. These differences may even allow for correct execution when the interpretation is constant throughout the enterprise. However, when these different interpretations exist in the same enterprise, errors occur. When the problem domain consists of millions of objects and thousands of object servers, the only feasible solution is to code generate the object servers.

Experimentation

By allowing a system engineer to try different service implementations and middleware without having to encode all of the possible combinations by hand, a system engineer can develop prototypes of multiple test architectures and evaluate their characteristics in realistic deployment environments.

One DARPA project ran into trouble when the deployment environment proved to be less reliable than it was assumed to be. The project used hand-held computers networked with radio waves. When the connections between the hand-held computers proved unreliable, the system performance was severely impacted. Basically, the system would connect to the object servers only to be disconnected by unreliable communications within minutes. The system spent most of its resources establishing and re-establishing connections. The project was able to move from a connection-based architecture using CORBA to a connectionless architecture using HyperText Transfer Protocol (HTTP)/XML by regeneration of the object servers with different templates.

Technology Evolution

By building your data access and dissemination layer for the enterprise with Quava, your enterprise architecture can handle changes in software technology more readily. When advanced implementations of core services become available, a new template that implements the glue code between the new service implementation and the objects is created, and the object servers are regenerated without having to change any client application software. Also, when new middleware technologies arise, the object servers can be regenerated with additional interfaces so that the object servers can support client applications using the previous interfaces and new client applications using the new interface simultaneously. Older interfaces can be removed when client applications no longer need them by regenerating the object servers without the deprecated interface.

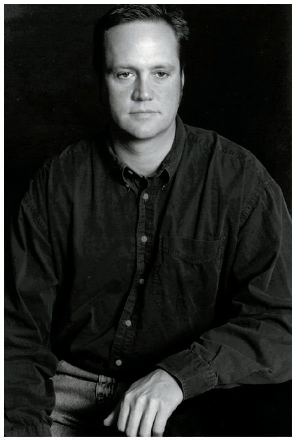
CONCLUSION

In our research, we found that model-driven code generation was a very promising technology with many benefits to the software practitioner. The benefits of using this approach in an enterprise help elevate many of the more substantial problems faced when developing large-scale systems. The openness and flexibility of the Quava implementation gives great support to life-cycle maintenance and software evolution of the system.

REFERENCES

1. Siska, C. 1998. "A Processor Description Language Supporting Retargetable Multi-Pipeline DSP Program Development Tools," *Proceedings on 11th International Symposium on System Synthesis*, 2–4 December, Taiwan, China, pp. 31–36
2. Bringmann, O., W. Rosenstiel, and D. Reichardt. 1998. "Synchronization Detection for Multi-Process Hierarchical Synthesis," *Proceedings on 11th International Symposium on System Synthesis*, 2–4 December, Taiwan, China, pp. 105–110.
3. Leone, M. and P. Lee. 1994. "Lightweight Run-Time Code Generation," *Proceedings of the ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation*, June.
4. Engler, D. 1996. "VCODE: A Retargetable, Extensible, Very Fast Dynamic Code Generation System," *Proceedings of the 23rd Annual ACM Conference on Programming Language Design and Implementation*, 21–24 May, Philadelphia, PA, pp. 160–170.
5. Berzins, V., O. Ibrahim, and Luqi. 1997. "A Requirements Evolution Model for Computer-Aided Prototyping," *Proceedings of the 9th International Conference on Software Engineering and Knowledge Engineering*, Madrid, Spain, June.
6. Shing, M., V. Berzins, and Luqi. 1996. "Computer-Aided Prototyping System (CAPS)," *Proceedings of the Software Technology Conference*, Salt Lake City, UT, April.
7. Attie, P. and E. Emerson. 1989. "Synthesis of Concurrent Systems with Many Similar Processes," *Proceedings of the 16th Annual ACM Symposium on Principles of Programming Languages*, 11–13 January, Austin, TX, pp. 191–201.
8. Lewis, J., S. Henry, D. Kafura, and R. Schulman. 1991. "An Empirical Study of the Object-Oriented Paradigm and Software Reuse," *Conference Proceedings on Object-Oriented Programming Systems, Languages, and Applications*, 6–11 October, Phoenix, AZ, pp. 184–196.
9. Bhansali, S. 1995. "A Hybrid Approach to Software Reuse," *Proceedings of the 17th International Conference on Software Engineering Symposium on Software Reusability*, 29–30 April, Seattle, WA, pp. 215–218.
10. Henderson-Sellers, B. and J. Edwards. 1990. "Object-Oriented Systems Life Cycle," *Communications of the ACM*, vol. 33, no. 9, pp. 142–159.

❖



William J. Ray

MS in Software Engineering,
Naval Postgraduate School,
1997

Current Research: Enterprise architectures; distributed systems; object-oriented technologies.

Andy Farrar

BS in Computer Science,
San Diego State University,
1992

Current Research: Middleware technologies; software synthesis; distributed systems.

CINC 21 Advanced Concept Technology Demonstration

Richard N. Griffin
SSC San Diego

ABSTRACT

This paper describes an Advanced Concept Technology Demonstration (ACTD) entitled Commander-in-Chief for the 21st Century (CINC 21) and documents the involvement of SSC San Diego personnel in the ACTD. The goal of the ACTD is to create a highly visual, dynamically updated capability to develop and understand the CINC's theater situation, plans, and execution status during multiple, simultaneous crises involving joint, coalition, and humanitarian agencies based on shared knowledge and collaboration across secure and optimized networks. The paper describes operational needs and focuses on the application of technologies in specific areas. CINC 21 is a 5-year program consisting of 3 years of "development and integration" and 2 years of "residual support and transition."

OVERVIEW

Commander-in-Chief for the 21st Century (CINC 21) was a Fiscal Year 2000 (FY 00) new-start Advanced Concept Technology Demonstration (ACTD). The Joint Requirements Oversight Council (JROC) approved CINC 21 on 11 February 2000, and Congressional approval followed on 13 March 2000. The program consists of 3 years of development and integration and 2 years of residual support and transition.

CINC 21's mission is to develop and assess new command and control concepts for improving the speed and effectiveness of joint, coalition, and inter-agency operations by leveraging advances in visualization, knowledge management, information management, and network technologies.

CINC 21 directly addresses the emphasis that Joint Vision 2010 (JV 2010) places on Information Superiority and Decision Superiority. JV 2010 describes Information Superiority as the ability to collect, process, and disseminate an uninterrupted flow of information while exploiting or denying adversaries' ability to do the same. However, Information Superiority, while necessary, is not sufficient. Success in any operation requires the ability to effectively use and quickly exploit information. JV 2020 refers to this ability as Decision Superiority and states "… creation of information superiority is not an end in itself…we have a competitive advantage only when it is effectively translated into superior knowledge and decisions."

Specifically, CINC 21 will address Decision Superiority by: improving the ability of the CINC's "extended" staff to track and manage multiple, simultaneous crises; enabling synchronized understanding of operations between CINCs and the commanders of joint task forces; instituting enhancements to the information infrastructure to match operational needs (including coalition and interagency needs); and increasing the speed of command decision-making to gain and maintain the strategic advantage. Table 1 lists the lead organizations executing CINC 21.

The CINC 21 team concluded the first year of the development program in October 2000 by conducting a successful major demonstration at United States Pacific Command (USPACOM) headquarters. The primary audience for the demonstration included the directors of each staff component; the Deputy CINC, LT Gen Case; and the Deputy Chief of Staff, MG Lowe. Based on the success of this demonstration, the development team adopted a model of delivering technology, configured in operational packages, in a development cycle of 4-month increments. Spiral I was approved in February 2001, with delivery of the technology scheduled for mid-May 2001. The first major Military Utility Assessment (MUA) opportunity for CINC 21 ACTD delivered under the spiral development process will take place during Exercise KERNEL BLITZ (Experimental) (KB (X)) in June 2001. Subsequent development spiral deliveries will occur in September 2001, January 2002, and May 2002. Other MUA events are currently undetermined, but the culminating "graduation event" will be scheduled for a USPACOM Exercise in the fourth quarter of FY 02.

TABLE 1. Participating organizations.

Deputy Under Secretary of Defense for Advanced Systems and Concepts (DUSD [AS & C])	ACTD Oversight Dr. Robert Popp
U.S. Pacific Command (USPACOM)	Operational Manager Mr. Randall Cieslak
Office of Naval Research	Technical Manager Dr. Sue Hearold
Defense Information Systems Agency	Deputy Technical Manager LTC Riki Barbour
Space and Naval Warfare Systems Command (SPAWAR)	Transition Manager Mr. John Quintana

FY 03 and FY 04 are transition years consisting of three major activities:

1. Providing operations and maintenance (O&M) support for leave behind/residual capabilities,
2. Continuing transition planning with acquisition sponsors and programs of record,
3. Continuing assessments for the Defense Information Infrastructure Common Operating Environment (DII COE) and modifying applications as necessary to meet compliance requirements.

Objectives

Detailed objectives for the ACTD are stated in the CINC 21 Implementation Directive. U.S. Commander in Chief, U.S. Pacific Command (USCINCPAC) defined Critical Operational Issues (COIs). Table 2 shows the relationship between the COIs and the CINC 21 objectives.

TABLE 2. Objectives and critical operational issues.

Objectives	Critical Operational Issues
· Improve situational awareness and understanding through a) shared understanding of operational situation, b) scaleable and tailorable visualization, c) advanced decision support and knowledge management tools.	· Can advanced visualization technology empower individuals to process, digest, and assimilate large volumes of information, thereby enabling faster, more effective decisions? · Can knowledge management technology integrate information, context, and rules to increase understanding and, therefore, improve decision-making?
· Demonstrate and synchronize distributed decision-making, collaboration, and information management/information dissemination tools among joint, coalition, inter-agency, and non-governmental organization partners.	· Can collaboration tools be used to overcome the tyrannies of time, distance, and system disparity?
· Enable command of the information enterprise through advanced enterprise management tools and user-specified and prioritized operational products.	· Can the collection of networks, databases, and applications be enhanced to optimize the flow of information, with security assurance, across multiple network enclaves?

Concept of Operations (CONOPS)

At the center of CINC 21's Concept of Operations (CONOPS) is a knowledge-enabled information sphere with tools and applications that will improve situational awareness and understanding, provide the ability to collaborate as necessary, and manage the information enterprise while transforming and accelerating the decision processes that support the management of crisis-contingency operations, the CINC's theater engagement policy, and supporting staff processes.

The CONOPS for crisis operations includes expanding the ability of warfighting CINCs to handle multiple crises by delegating planning and execution to distributed crisis management cells and by simplifying the information flow to CINC and Commander, Joint Task Force (CJTF) decision cells. A combination of intelligent information management and continuous collaboration with multiple crisis cells will accomplish this task. Benefits will accrue to the CINC headquarters, supporting and supported CINCs, subordinate unified commands, Department of Defense (DoD) and non-DoD agencies, non-government organizations, and coalition partners. CINC 21 addresses the need for CINCs and CJTFs to operate in this complex world environment by exploiting the power of visualization to convey knowledge and understanding.

In addition to traditional military operations, the 21st century environment makes it necessary to participate in a wide variety of theater engagement activities. These mission areas, sometimes known as Military Operations-Other-Than-War (MOOTW), include refugee control, humanitarian assistance, disaster relief, non-combatant evacuation, public security/law and order, support to host governments, mediation/negotiations, and demilitarization operations. All these operations put a premium on open-source/unclassified information that can be readily shared with all participants.

The desired outcome for this environment is threefold: (1) the necessary mature and maturing tools will be integrated to enable open-source information to be added to the information-gathering systems available to the USPACOM virtual staff, (2) the information will be compatible with decision-support software tools that enable assessment, evaluation, and prioritization of appropriate courses of action (COAs), and (3) the open-source information should be accessible from a mobile or remote command site/"cell."

As Figure 1 shows, CINC 21 will provide a highly visual, dynamically updated capability to develop and understand the CINC's theater situation, plans, and execution status during multiple, simultaneous crises involving joint, coalition, and humanitarian agencies based on shared knowledge and collaboration across secure and optimized networks. CINC 21 will provide the following capabilities:

- User-tailorable, integrated situation display
- Enhanced visualization of information so decision-makers can quickly interpret, assimilate, and act
- Secure access to relevant information at its source on demand (demand can be from user or intelligent agent)
- Distributed collaborative environment enabling rapid command and control and access to expertise at its source—collaboration as a basic service
- Enhanced security by providing capability to establish trusted network relationships on demand
- Information flow monitoring and dynamic allocation of resources to optimize distribution of information based on commanders' priorities

Technical Approach

CINC 21 seeks to provide an enhanced decision support environment for the CINC and its extended staff through mature commercial and government software packages. The objectives of the CINC 21 ACTD can be mapped into the four technical areas listed in Table 3.

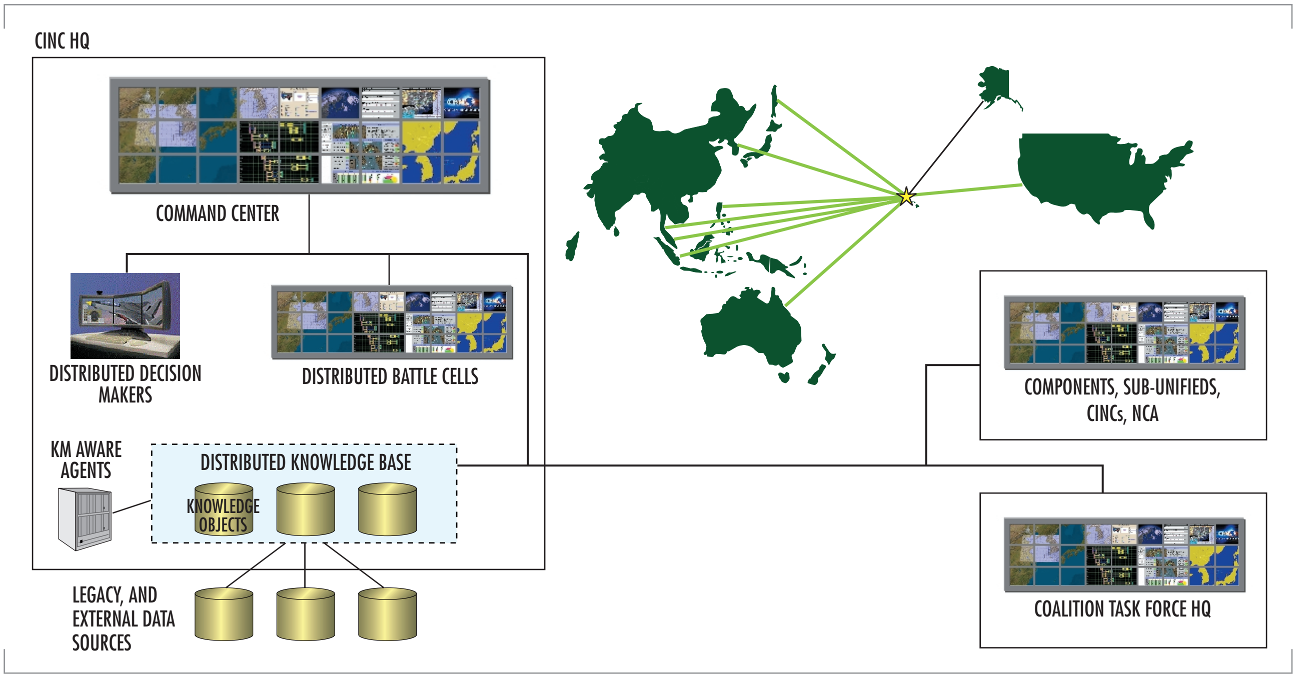

CINC HQ
COMMAND CENTER
DISTRIBUTED DECISION MAKERS
DISTRIBUTED BATTLE CELLS
KM AWARE AGENTS
DISTRIBUTED KNOWLEDGE BASE
KNOWLEDGE OBJECTS
LEGACY, AND EXTERNAL DATA SOURCES
COMPONENTS, SUB-UNIFIEDS, CINCs, NCA
COALITION TASK FORCE HQ

FIGURE 1. CINC 21 operational concept.

System Design, Engineering, and Integration

The CINC 21 responsibility for system design, engineering, and integration is assigned to the System Development Team led by Ray Glass (SSC San Diego). This team is responsible for all development activities leading up to the hand-off of a robust, configuration-managed hardware/software solution to the CINC 21 Implementation Team, led by Tom Tiernan, SSC San Diego.

The breakdown of the CINC 21 development activities into the four areas has been done to carefully split the responsibilities of the Development Team so that they can concentrate more fully on their primary objectives. Designers of the general framework services will not be inclined to shortcuts because of pressure in delivering specific operational packages. Operational package developers will be freed from the responsibilities of building and maintaining the core services. Integration, test, and configuration management has been separated from both activities to ensure unbiased independent verification and validation (IV&V). Finally, the operational support activity has been called out separately to

TABLE 3. CINC 21 technical areas.

Technical Area	Description
Data interoperability (information management, knowledge management, network infrastructure)	Provide improved mechanisms for sharing information across the CINC's staff and to enable more effective and efficient production of cross-staff decision products.
Information infrastructure enhancements (information management, knowledge management)	Provide upgrades to the CINC's information infrastructure that improve decision-making, foster greater inter-agency and coalition interaction, and improve security.
Knowledge wall environment (visualization)	Provide a structured environment that allows the rapid development and easy sharing of a wide range of correlation, visualization, and collaboration services. As an adjunct to this activity, CINC 21 will pursue the delivery of multi-panel desktop and wall-based displays as residual capabilities.
Operational packages (knowledge management)	Develop specific operational capabilities for USPACOM and United States Strategic Command (USSTRATCOM) by using a complete set of Extensible Markup Language (XML), Decision Tagged Data (DTD), databases, correlation, and visualization plug-ins to create useful end-to-end services.

ensure that support to the Implementation Team does not have a resource impact on other system development activities.

To ensure a common foundation for all three classes of CINC 21 users, the system development activity will be divided into three parts: a system design effort that will design and develop the user-independent CINC 21 foundation, an operational package development and operational support activity effort that will provide domain-specific products to operational users, and an implementation management effort responsible for integrating the operational packages into operational use. Figure 2 shows the system development approach.

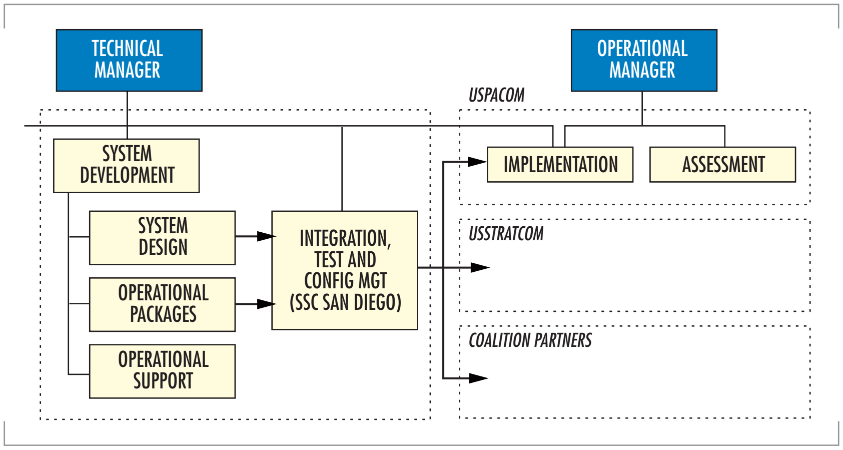


FIGURE 2. System development approach.

CONCLUSION

As stated in the introduction to this paper, CINC 21's objective is to increase the speed of command across the spectrum of operations by controlling and exploiting an information-rich environment. This objective demands advanced technologies linked to advanced concepts. Within CINC 21, we are exploring concepts and technologies that not only improve the ability to collect, process, and disseminate information, but also fundamentally change the way warfighters use that information by applying tools and processes that create knowledge and understanding. Today, we drown people in information, but leave them starving for knowledge. With CINC 21, we will show how we can significantly improve the ability to command and control forces by providing a more visual, structured, and interactive command environment.

ACKNOWLEDGMENTS

SSC San Diego personnel played key roles in the development of operational requirements for advanced technology and in shepherding the ACTD proposal through the approval process. They continue to play major roles in the management of the ACTD and in the integration of technological solutions to apply to warfighter requirements. Jeff Grossman, Tom Tiernan, and Dick Griffin were major players in the development of concepts and requirements. Sue Hearold (on loan to the Office of Naval Research [ONR]) is the Technical Manager, Ray Glass is the System Development Manager, Tom Tiernan is Systems Implementer, Pete Wussow leads the development of decision-focused visualization tools, and Dick Griffin is Deputy Operational Manager and leads the military utility assessments phase. Mike Reilley, USCINCPAC Science and Technology Advisor, provides oversight and liaison with technology developers.

❖

Richard N. Griffin
MA in International Studies, The Johns Hopkins University, 1977
Current Work: Deputy CINC 21 Operational Manager, U.S. Pacific Command.